Statistical Techniques in Robotics (16-831, S21) Lecture #13 (Wednesday, March 17)

Thompson Sampling

Lecturer: Kris Kitani *Scribes: Abhinav Agarwalla, Kshitij Goel*

1 Review

In the previous lecture, we finished learning about the Explore-Exploit algorithm and the Upper Confidence Bound (UCB) algorithm for the Multi-Armed Bandit (MAB) problem.

1.1 Explore-Exploit Algorithm

The algorithm for the Explore-Exploit algorithm is shown in Alg. 1.

Algorithm 1 Explore-Exploit

```
for k = 1 → K do
    for m = 1 → M do
        a = k
        Receive(r)
        μ̂_k = μ̂_k + r/M
    end for
end for
for t = KM → T do
    a^(t) = arg max_k' μ̂_k'
    Receive(r^(t))
end for
```

We use the Hoeffding's inequality in the regret bound derivation for the Explore-Exploit algorithm. The regret bound derivation proceeds in two phases: (1) Explore phase and (2) Exploit phase. For the explore phase, we get the regret bound to be $R_{\text{explore}} = \mathcal{O}(KM)$ and for the exploit phase it came out to be $R_{\text{exploit}} = \sum_{t=KM+1}^{T}(\mu_{k^*}^{(t)} - \mu_{\hat{k}}^{(t)}) \leq (T - KM) \cdot 2\sqrt{\frac{\log(2/\delta)}{2M}}$. Combining these phase bounds, we get the overall regret bound of the Explore-Exploit algorithm to be:

$$\begin{aligned} R_{\text{explore-exploit}} &= R_{\text{explore}} + R_{\text{exploit}} \\ &= KM + (2T - KM) \cdot \sqrt{\frac{1}{M}} \\ &\leq KM + 2T \cdot \sqrt{\frac{1}{M}} \end{aligned}$$

Optimal M can be computed by differentiating the RHS of the above inequality. It comes out to be $M = \left(\frac{T}{K}\right)^{2/3}$. Substituting this back into the expression for the overall bound, we get the final regret bound to be $R_{\text{explore-exploit}} = \mathcal{O}(K^{1/3}T^{2/3})$. Note that the grow in regret is sub-linear with respect to time. Therefore, the Explore-Exploit algorithm is a no-regret algorithm.

1.2 Upper Confidence Bound (UCB) Algorithm

The confidence term is obtained using Hoeffding's inequality and depends on the number of pulls of a particular arm $T_{k'}^{(t)}$, total pulls T and δ. So, as the game progresses and the number of pulls increase, the learner becomes more confident and the confidence term reduces.

Algorithm 2 Upper Confidence Bound (UCB)

for $t = 1 \to T$ **do**
  if $t \leq K$ **then**
    $k = t$ ▷ Initially pull each arm once (exploration)
  else
    $k =_{k'} \left(\hat{\mu}_{k'} + \sqrt{\frac{\log(2T/\delta')}{2T_{k'}^{(t)}}} \right)$ ▷ upper confidence
  end if
  `RECEIVE`$(r^{(t)})$
  $T_k^{(t)} = T_k^{(t')} + 1$ ▷ update pull counter
  $\hat{\mu}_k = \frac{1}{T_k^{(t)}} \left(\hat{\mu}_k (T_k^{(t)} - 1) + r^{(t)} \right)$ ▷ update mean reward for k
end for

For UCB, the regret bound comes out to $\mathcal{O}(\sqrt{KT})$. In this case also the regret grows sub-linearly with respect to time – the UCB algorithm is also no-regret.

2 Summary

Definition 1. Bayesian Stochastic Bandit Each bandit is assumed to have a generative distribution from which each reward is sampled. Thus, $r \sim p(r|a, \theta)$ where r, a, θ denote reward, action and the parameter for generative distribution.

2.1 Thompson Sampling

Thomson Sampling requires assuming a Bayesian Stochastic Bandit. In other words, it assumes that the reward is generated from a distribution which is parameterized by θ. Since θ isn't directly observable and hence unknown, it maintains a running estimate of θ, denoted by $\hat{\theta}$, by observing the rewards. To select the arm to pull, we select the arm with the highest expected reward. Mathematically:

$$a = \arg\max_k \mathbb{E}_{p(r|a_k, \hat{\theta}_k)} \left[r | a_k, \hat{\theta}_k \right]$$

In case the actual θ^* was known, we could simply replace $\hat{\theta}$ with θ^* in the above equation.

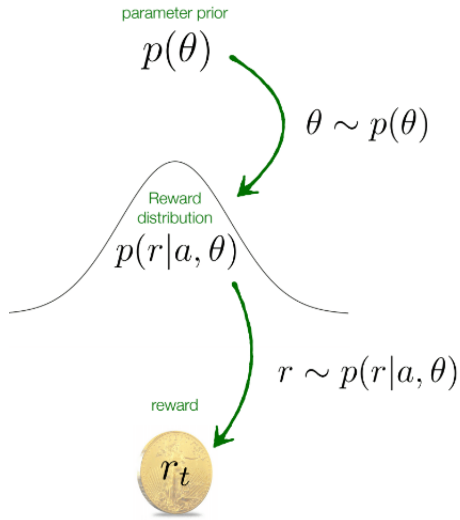

Figure 1: Bayesian Stochastic Bandits

2.1.1 Estimating θ

To estimate the true parameter θ, we would use a history of actions taken and reward received to condition the parameter. The argument is very similar to a maximum likelihood (MLE) argument, or strictly maximum-a-posteriori (MAP) in our case. We wish to obtain an estimate $\hat{\theta}$ that maximizes the likelihood of observing the history of actions and rewards. Mathematically,

$$p(\theta|h^{(t)}) = p(\theta|a^{(1)}, r^{(1)}, \ldots, a^{(t)}, r^{(t)})$$
$$\hat{\theta} = \arg\max_{\theta} p(\theta|h^{(t)})$$

where $h^{(t)} = \{a^{(1)}, r^{(1)}, \ldots, a^{(t)}, r^{(t)}\}$ is the history of actions and rewards.

Now that we know how we are going to get the estimate for θ. Lets see how we can compute the estimate $\hat{\theta}$.

Using Bayes rule:

$$p(\theta|a^{(1)}, r^{(1)}, \ldots, a^{(t)}, r^{(t)}) = \frac{p(r^{(1)}, \ldots, r^{(t)}|\theta, a^{(1)}, \ldots, a^{(t)})p(\theta|a^{(1)}, \ldots, a^{(t)})}{p(r^{(1)}, \ldots, r^{(t)}|a^{(1)}, \ldots, a^{(t)})}$$

Now in a bandit setting, the next state is independent of the actions taken by a learner. In other words, θ is independent of all $a^{(t)}$'s.

$$p(\theta|a^{(1)}, r^{(1)}, \ldots, a^{(t)}, r^{(t)}) = \frac{p(r^{(1)}, \ldots, r^{(t)}|\theta, a^{(1)}, \ldots, a^{(t)})p(\theta)}{p(r^{(1)}, \ldots, r^{(t)}|a^{(1)}, \ldots, a^{(t)})}$$

Assuming that the reward $r^{(t)}$ is conditionally independent of other rewards given the action and the parameter θ, ie. rewards are i.i.d. We have,

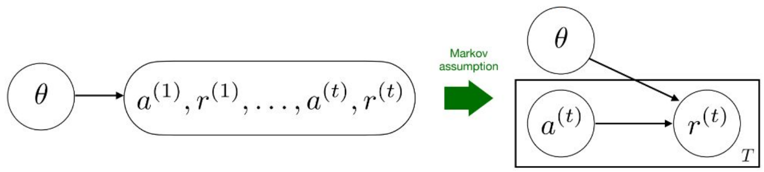

Figure 2: $r^{(t)}$ is dependent on $a^{(t)}$ and θ, while $a^{(t)}$ and θ are independent

$$p(\theta|a^{(1)}, r^{(1)}, \ldots, a^{(t)}, r^{(t)}) = \frac{\prod_t p(r^{(t)}|\theta, a^{(1)}, \ldots, a^{(t)})p(\theta)}{\prod_t p(r^{(t)}|a^{(1)}, \ldots, a^{(t)})}$$

Now, using Markov assumption $p(r^{(t)}|\theta, a^{(1)}, \ldots, a^{(t)} = p(r^{(t)}|\theta, a^{(t)})$:

$$p(\theta|a^{(1)}, r^{(1)}, \ldots, a^{(t)}, r^{(t)}) = \frac{\prod_t p(r^{(t)}|\theta, a^{(t)}p(\theta)}{\prod_t p(r^{(t)}|a^{(t)})}$$

Thus the posterior distribution simplifies to:

$$p(\theta|h^{(t)}) \propto \prod_t p(r^{(t)}|\theta, a^{(t)}p(\theta)$$

which can be written in an incremental fashion:

$$p(\theta|h^{(t)}) \propto p(r^{(t)}|\theta, a^{(t)}p(\theta|h^{(t-1)})$$

Now that we have an incremental update to the posterior $p(\theta|h^{(t)}$, we use this to obtain our estimate $\hat{\theta}_k$ for the k^{th} arm as:

$$\hat{\theta}_k = \arg\max_{\theta_k} p(\theta_k|h_k^{(t)})$$

$$\hat{\theta}_k = \arg\max_{\theta_k} \underbrace{p(r^{(t)}|a_k^{(t)}, \theta_k)}_{\text{likelihood}} \underbrace{p(\theta_k|h_k^{t-1})}_{\text{prior}}$$

In the above equation, we can replace $r^{(t)}$ with $r_k^{(t)}$ which would mean the same thing. Here, it's implicit that the reward $r^{(t)}$ is obtained after picking arm k.

We can greatly simplify the complex posterior updates by assuming certain distributions for prior and likelihood, which is covered in the next section.

2.1.2 Conjugate Priors

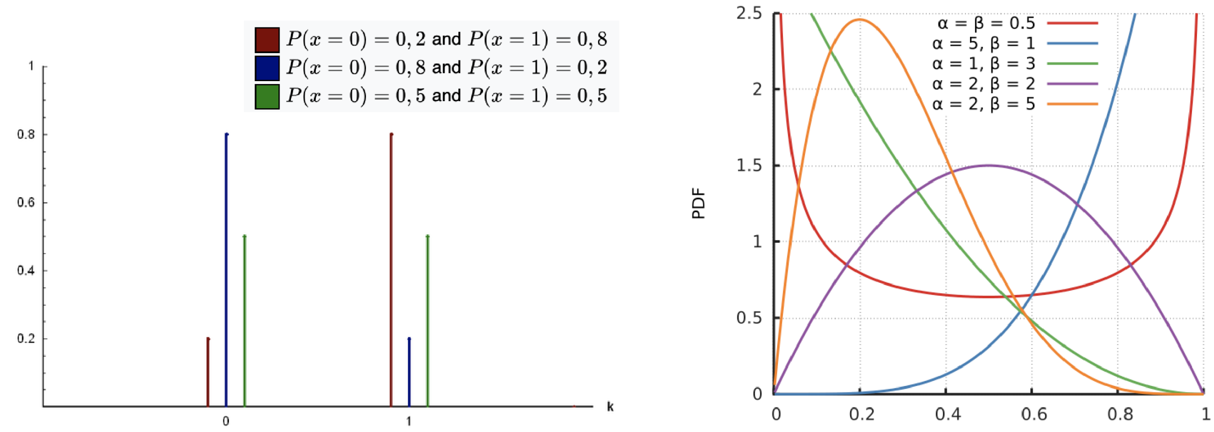

Figure 3: Conjugate Priors. Beta distribution (right) is a conjugate prior of the Bernoulli distribution (left).

Consider the general posterior estimation scenario:

$$\underbrace{p(\theta \mid x)}_{\text{posterior}} \propto \underbrace{p(x \mid \theta)}_{\text{likelihood}} \underbrace{p(\theta)}_{\text{prior}}$$

When the posterior and the prior have the same type of distribution, they are called *conjugate distributions*. In this case, the prior is called a *conjugate prior*. For example, the Beta distribution is the conjugate prior of the Bernoulli distribution:

$$p(r \mid \theta) = \theta^r (1-\theta)^{1-r} \qquad \text{(Bernoulli Distribution)}$$

$$p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1} \qquad \text{(Beta Distribution)}$$

where $\Gamma(n) = (n-1)!$ is the Gamma function. We can easily show that posterior is Beta distribution if the *likelihood* is a Bernoulli distribution and *prior* is a Beta distribution.

$$\begin{aligned} p(\theta \mid r) &\propto p(r \mid \theta) p(\theta) \\ &\propto \theta^r (1-\theta)^{1-r} \theta^{\alpha-1} (1-\theta)^{\beta-1} \\ &\propto \theta^{(r+\alpha-1)} (1-\theta)^{(1-r+\beta)-1} \\ &\propto \theta^{(\alpha'-1)} (1-\theta)^{\beta'-1} \end{aligned}$$

We observe that the posterior can be calculated efficiently via additive updates:

$$\begin{aligned} \beta' &= \beta + 1 - r \\ \alpha' &= \alpha + r. \end{aligned}$$

Thus, we can use the conjugate distributions to design an efficient strategy to update the *maximum-a-posteriori* estimate for θ. We now study the generic algorithm for Thompson Sampling followed by a specific example that uses Beta conjugate prior for Bernoulli likelihood.

2.1.3 Thompson Sampling Algorithm: Generic and Bern-Beta

The generic algorithm for Thompson Sampling [2] is shown in Alg. 3.

Algorithm 3 Thompson Sampling (Incremental)

for $t = 1 \to T$ **do**
    $\theta_k \sim p(\theta_k \mid h_k)$ ▷ sample from posterior
    $a_{\hat{k}}^{(t)} = \arg\max_k \mathbb{E}_{p(r|a_k,\theta_k)}[r \mid a_k, \theta_k]$ ▷ predict
    Receive($r^{(t)}$) ▷ get sampled reward
    $p(\theta_{\hat{k}} \mid h_{\hat{k}}) \propto p(r^{(t)} \mid a_{\hat{k}}^{(t)}, \theta_{\hat{k}} p(\theta_{\hat{k}} \mid h_{\hat{k}}))$ ▷ update posterior
end for

Let us understand this generic algorithm with the Bernoulli-Beta case. The algorithm proceeds as shown in Alg. 4.

Algorithm 4 Bern-Beta Thompson Sampling

for $t = 1 \to T$ **do**
    $\theta_k \sim p(\theta_k; \alpha_k, \beta_k)$ ▷ sample from posterior
    $a_{\hat{k}}^{(t)} = \arg\max_k \mathbb{E}_{p(r|a_k,\theta_k)}[r \mid a_k, \theta_k]$ ▷ predict
    Receive($r^{(t)}$) ▷ get sampled reward
    $\alpha_{\hat{k}} = \alpha_{\hat{k}} + r^{(t)}$ ▷ update posterior
    $\beta_{\hat{k}} = \beta_{\hat{k}} + 1 - r^{(t)}$ ▷ update posterior
end for

For each time step t, first a parameter estimate θ_k is sampled from the posterior $p(\theta_k; \alpha_k, \beta_k)$ for all arms $k = \{1, \ldots, K\}$. Then, the action $a_{\hat{k}}^{(t)}$ is picked that maximized the expected reward over the arms. After pulling the arm we get some reward at this time step, $r^{(t)}$. Using this reward, the posterior is updated by changing $\alpha_{\hat{k}}$ and $\beta_{\hat{k}}$ for the arm that was pulled. The process repeats for all the time steps.

2.1.4 Empirical Performance Comparison with UCB

Empirically, [1] provides an empirical performance comparison of the Thompson Sampling algorithm with respect to the UCB algorithm (Fig. 4). It is observed that Thompson sampling has a lower regret than UCB, especially when the timesteps is large. This effect is true for different values of the number of arms K, and the ϵ.

Interestingly, it performs better than the asymptotic lower bound:

$$\mathcal{R}(T) \geq \log(T) \left[\sum_{i=1}^{K} \frac{p^* - p_i}{D(p_i || p^*)} + O(1) \right]$$

Theoretically, the regret for Thompson Sampling is known to be $\mathcal{R}(T) = \mathcal{O}(\sqrt{KT \log T})$.

where $p^* = \max p_i$ and $D(p_i || p^*)$ is the KL-divergence between p_i and p^*.

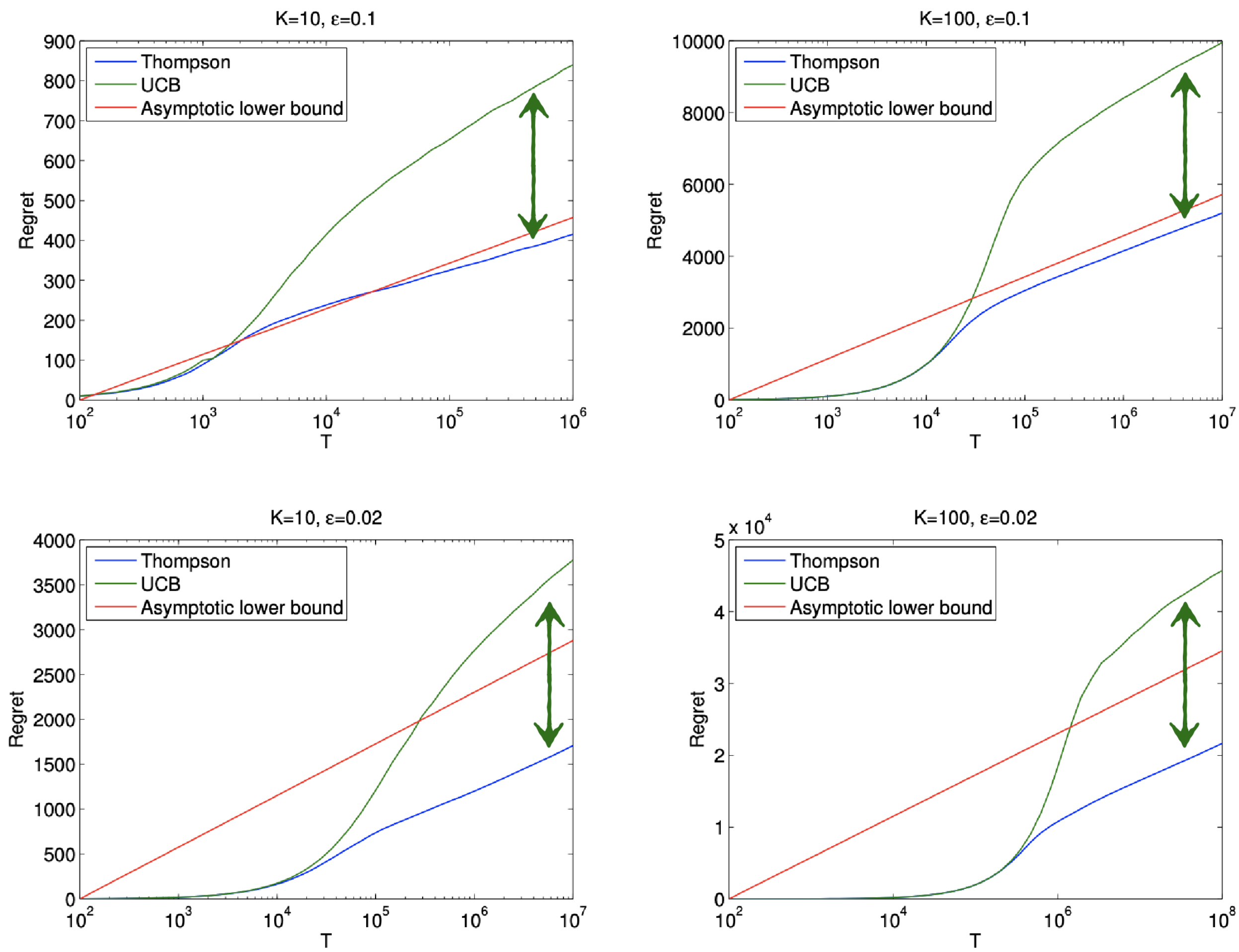

Figure 4: Empirical performance comparison between Thompson Sampling and the UCB algorithms [1].

References

[1] O. Chapelle and L. Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24:2249–2257, 2011.

[2] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.